primary

US-PAT-NO: **6038563**

DOCUMENT-IDENTIFIER: US 6038563 A

TITLE: System and method for restricting database access to managed object information using a permissions table that specifies access rights corresponding to user access rights to the managed objects

DATE-ISSUED: March 14, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bapat; Subodh	Palo Alto	CA	N/A	N/A
Fisher; Bart Lee	Sunnyvale	CA	N/A	N/A

US-CL-CURRENT: 707/10, 707/203 , 707/8 , 707/9

ABSTRACT:

An access control database has access control objects that collectively store information that specifies access rights by users to specified sets of the managed objects. The specified access rights include access rights to obtain management information from the network. An access control server provides users access to the managed objects in accordance with the access rights specified by the access control database. An information transfer mechanism sends management information from the network to a database management system (DBMS) for storage in a set of database tables. Each database table stores management information for a corresponding class of managed objects. An access control procedure limits access to the management information stored in the database tables using at least one permissions table. A permissions table defines a subset of rows in the database tables that are accessible to at least one of the users. The set of database table rows that are accessible corresponds to the managed object access rights specified by the access control database. A user access request to access management

information in the database is intercepted, and the access control procedure is
invoked when the user access request is a select statement. The database
access engine accesses information in the set of database tables using the
permissions tables such that each user is allowed access only to management
information in the set of database tables that the user would be allowed by the
access control database to access.

13 Claims, 25 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 17

---------- KWIC ---------

US Patent No. - PN (1):
6038563

Brief Summary Text - BSTX (16):
Another issue not addressed by X.741 is that customers of large networks
often insist upon the ability to generate network management reports using
"SQL" type report generators. That is, users of such networks want or require
the ability to generate reports on the status of their network resources, while
avoiding the complexities of network management information retrieval using
SNMP (or any other network management protocol). X.741 and the related
standards do not call for, or even suggest, any type of direct SQL-type access
to the **managed object database** for the purpose of generating management
reports. In fact, direct SQL-type access might be seen as contrary to the
goals of X.741 since it is a potential source of **security** leaks.

Drawing Description Text - DRTX (10):
FIG. 8 depicts the event **registry** and event router portions of a management
information server in a preferred embodiment of the present invention.

Detailed Description Text - DETX (22):
In particular, the MIS 150 only performs access control for objects at the
top of the managed objects tree, while each of the auxiliary servers

performs
access control for objects in respective designated subtrees of the managed
objects tree. One important exception to the above statement is that all
access requests for event notifications (i.e., with an operation of "receive
notification from") are delivered to an event **registry** module in the MIS,
regardless of which objects are the targets of the access request. This is
discussed in more detail below with respect to event notification access
control.

Detailed Description Text - DETX (41):
an event **registry** 184, which is a mechanism for keeping track of event
notifications that particular users have requested; and

Detailed Description Text - DETX (91):
Referring to FIG. 6, there is shown the sequence of actions performed by the
access request partitioning and routing procedure 172, the access control
decision and enforcement functions 176, 174, and the request response combining
procedure 178. Note that this discussion does not apply to access requests for
event notifications, which are handled separately by the event **registry**.

Detailed Description Text - DETX (110):
Referring to FIG. 8, the MIS 150 maintains an event **registry** 184. More
accurately, the event **registry** 184 is a software module that maintains a table
260 of user event requests. The MIS directs all access requests whose
specified operation type is "event notification" to the event **registry** 184,
regardless of which objects are specified by the request. The table 260 stores
information denoting, for specified event notifications that can be generated
by either the managed objects or the access control objects, which users or
other entities have registered a requested to receive copies of those event
notifications. The event **registry** table 260 only stores information about
events that users and other entities have requested. The event notification
registration requests (which are access requests with an operation type

equal
to "event notification") can be specified either in terms of specified objects,
specified classes of objects, or specified subtrees of objects. Thus, for
instance, a user could request receipt of all event notifications for router
objects (i.e., which is a class of objects), and could further specify a
filter, such as only routers located in the state of California or routers
manufactured by a particular company. Users and entities can also revoke prior
requests.

Detailed Description Text - DETX (111):
In the preferred embodiment, the event **registry** 184 only checks registration
requests to ensure that the requests are semantically correct and that the
specified objects for which events are requested actually exist. Thus, in the
preferred embodiment the event **registry** 184 does not check to see if the user
or entity making a registration request has the security clearance to actually
receive the requested notifications. That job is given to the event router
186, which checks event notification access rights at the time each event
notification is being processed. As a result, any changes in a user's access
rights to event notifications are taken into account by the event router and do
not affect the information stored in the event **registry** 184.

Detailed Description Text - DETX (113):
All event notifications, including event notifications generated by managed
objects (indicated by "other event sources" in FIG. 8) and event notifications
generated by access control objects (indicated by the special auxiliary server
154 in FIG. 8), are delivered to the event router 186 in the MIS 150. The
event router 186 also has access to the access control tree 170 and the table
of user event requests 260 in the event **registry** 184. For each event
notification received by the event router 186, the event router first
determines which users and entities have requested a copy of that event
notification, and then determines which of those users and entities have the
right to receive those event notifications. The determination of access rights

to event notifications is performed using the access control decision function,
as shown in FIG. 5. Thus, the event router looks, in sequence, at the global
deny rule, the targeted deny rules, the global grant rule and the targeted
grant rules until a matching rule is identified. A default rule is applied if
no matching rule is found. A matching rule must (A) apply to the "event
notification" operation, (B) apply to the object that generated the event
notification, and (C) apply to a group of which the requester is a member.

Detailed Description Text - DETX (115):
One specific application of the event **registry** 184 and event router 186 used
in the preferred embodiments is as follows. There is a special auxiliary
server 154 that handles all access requests to and modifications of the access
control tree 170. In other words, access requests (other than event
notification access requests) whose target set is located in the access control
tree 170 are routed by the MIS 150 to the special auxiliary server 154.

Furthermore, all changes to the access control tree 170 result in the
generation of event notifications that are sent to the event router 186. In
particular, the creation of new access control objects, the deletion of access
control objects, and the modification of the attributes of any access control
object, all result in the generation of event notifications.

Detailed Description Text - DETX (116):
The MIS 150 and auxiliary servers 152 are all automatically registered in
the event **registry** 184 to receive all event notifications related to changes in
the access control tree 170. The MIS 150 and auxiliary servers are also
included in a set of "super users" with access rights to all event
notifications. Furthermore, among the library procedures shared by the MIS 150
and auxiliary servers 152 is an event receiving and processing procedure 262.
When the MIS 150 and auxiliary servers 152 receive any event notifications
indicating a change in the access control tree 170, the event processing
procedure 262, which is invoked by each server, makes the same change to the

server's local copy of the access control tree 170. As a result, the local
copies of the access control tree 170 in each of the servers 150, 152 are
updated virtually simultaneously.

Detailed Description Text - DETX (118):
X.741 does not call for, or even suggest, SQL access to the **managed object database**. In fact, direct access via a DBMS mechanism might be seen as contrary to the goals of X.741 since it is a potential source of **security** leaks. However, corporate customers of large communication networks are demanding direct "read only" access to management information for purposes of report generation.

Detailed Description Text - DETX (124):
The log server 290 is registered with the event **registry** to receive all event notifications generated by the system, and has corresponding access rights. The log server 290 is preferably a software entity or process that runs on the same computer or computer node as the MIS 150. A set of filters 291, 294 in the log server 290 determine which event notifications are stored, as well as where. A first filter 291 in the log server is called the security audit trail filter. This filter 291 passes "access grant" and "access denial" event notifications generated by the MIS 150 and auxiliary servers 152 (see FIG. 8). The security audit trail filter 291 can selectively store either the entire event notification, or a specified portion of it, in the security audit trail file 182. More specifically, when the security audit trail is configured to work in a detailed mode, the security audit trail 182 stores every access request and the corresponding outcome in its entirety. When the security audit trail is configured to work in an abbreviated mode, the security audit trail 182 stores a shortened representation of every access request and the corresponding outcome.

Detailed Description Text - DETX (128):
(B) defining and creating a database object 298, and registering the database object 298 with the event **registry** to receive event

notifications
affecting the rights of users to receive those event notifications; the database object 298 includes a first attribute that contains a list of the DBMS
tables in which the event log is stored, and a second attribute that contains a
list of the groups with access rights to the event notifications;

Detailed Description Text - DETX (131):
For each event log 282 there are one or more corresponding target objects in
the access control object tree 170 that define (1) the target set of managed
objects for which event notifications are to be stored in the event log, and
(2) the types of event notifications to be included in the event log. For any
particular event log, the set of groups of authorized users must be the same
for all event notifications in that event log. Any changes in the groups of
users to be granted access to the event log are communicated to the corresponding database object 198 by registering the database object with the
event **registry** to receive event notifications about attribute changes to the
target object(s) corresponding to the event log. The database object 298 is
also registered to receive event notifications of attribute changes to the
group objects for the groups that have access rights to the event log.

Detailed Description Text - DETX (148):
The FDN operates as the primary **key** to the data stored in the table. Using
security mechanisms that will be described below, the FDN is used as the **key**
that determines which managed objects that a particular user is permitted to
access or modify.

Secondary

US-PAT-NO: 6038563

DOCUMENT-IDENTIFIER: US 6038563 A

TITLE: System and method for restricting database access to managed object information using a permissions table that specifies access rights corresponding to user access rights to the managed objects

DATE-ISSUED: March 14, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bapat; Subodh	Palo Alto	CA	N/A	N/A
Fisher; Bart Lee	Sunnyvale	CA	N/A	N/A

US-CL-CURRENT: 707/10, 707/203 , 707/8 , 707/9

ABSTRACT:

An access control database has access control objects that collectively store information that specifies access rights by users to specified sets of the managed objects. The specified access rights include access rights to obtain management information from the network. An access control server provides users access to the managed objects in accordance with the access rights specified by the access control database. An information transfer mechanism sends management information from the network to a database management system (DBMS) for storage in a set of database tables. Each database table stores management information for a corresponding class of managed objects. An access control procedure limits access to the management information stored in the database tables using at least one permissions table. A permissions table defines a subset of rows in the database tables that are accessible to at least one of the users. The set of database table rows that are accessible corresponds to the managed object access rights specified by the access control database. A user access request to access management information in the database is intercepted, and the access control

procedure is
invoked when the user access request is a select statement. The database
access engine accesses information in the set of database tables using the
permissions tables such that each user is allowed access only to management
information in the set of database tables that the user would be allowed by the
access control database to access.

13 Claims, 25 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 17

---------- KWIC ---------

Brief Summary Text - BSTX (16):
Another issue not addressed by X.741 is that customers of large networks
often insist upon the ability to generate network management reports using
"SQL" type report generators. That is, users of such networks want or require
the ability to generate reports on the status of their network resources, while
avoiding the complexities of network management information retrieval using
SNMP (or any other network management protocol). X.741 and the related
standards do not call for, or even suggest, any type of direct SQL-type access
to the **managed object database** for the purpose of generating management
reports. In fact, direct SQL-type access might be seen as contrary to the
goals of X.741 since it is a potential source of **security** leaks.

Detailed Description Text - DETX (118):
X.741 does not call for, or even suggest, SQL access to the **managed object
database**. In fact, direct access via a DBMS mechanism might be seen as
contrary to the goals of X.741 since it is a potential source of **security**
leaks. However, corporate customers of large communication networks are
demanding direct "read only" access to management information for purposes of
report generation.

Detailed Description Text - DETX (161):
For a given user and database table, the Create.sub.-- View

procedure 362
creates a unique user target map called the userTargetMap. The userTargetMap
has a global Deny Flag, a global Grant Flag, an item **Deny list** and an item
Grant list. An item corresponds to a row and the row stores management
information associated with one managed object. To construct the userTarget
Map, the Create.sub.-- View procedure 362 checks the rules in the access
control tree 170 to determine if the given user is in fact an authorized user
with access to at least some managed objects. If so, then Create.sub.-- View
analyzes every rule applicable to both the given user and the objects in the
given table and updates the userTargetMap by either setting the appropriate
flag of the userTargetMap or by updating the list of the FDNs in the item **Deny
list** and/or item Grant list.

Detailed Description Text - DETX (165):
If the default access action is to grant access, or the global Grant Flag is
TRUE, then the Create.sub.-- View procedure 362 inspects the item **Deny List** and
adds to the Where clause the FDN of every row (item) in the applicable table
not included in the item **Deny List**.

Detailed Description Text - DETX (221):
In this third embodiment, access control for a particular user on a particular managed object is defined by a permissions table or tables 406.
Preferably, the present invention has an access Grant table 408 and an access
Deny table 410. Each table stores permission entries 412, 414.

Detailed Description Text - DETX (229):
The Grant table 408 stores a list of all access Grant permissions. The **Deny
table** 410 stores a list of all access Deny permissions. When checking whether
access should be permitted for a particular operation, the access control
procedure 404 checks both tables.

Detailed Description Text - DETX (234):
If the rule in the access control database specifies "global grant to user
U1 with item deny for items X1, X2 and X3" for operation type Op1, the

following entries are made in the grant table 408 and the **deny table** 410:

Detailed Description Text - DETX (235):
If the rule in the access control database specifies "deny user U1 access to all items except items X1, X2 and X3" for operation type Op1, then the following entries are made in the **deny table** 410 and grant table 408:

Detailed Description Text - DETX (236):
If the rule says "global deny" to user U1 for the operation type Op1, the following entry is made in the **deny table** 410:

Detailed Description Text - DETX (239):
The method described above is a more efficient way to store access control rules than storing only explicit grant rules or storing only explicit deny rules. For example, if one were to store only grant rules, then in a system with 5,000 managed objects, a new user given a global grant with a single item deny would require 4,999 records in the Grant Table 408. Using the method described above, the new user would have just two entries: one entry in the Grant Table and another entry in the **Deny table** 410.

Detailed Description Text - DETX (240):
Rules in the access control tree 170 that are defined in terms of a scope and filter are evaluated before entries are made in the grant table 408 and **deny table** 410. For example, if a scope and filter on the managed object tables results in a set of ten managed objects to which access must be granted with the rest being globally denied, then ten entries are made in the grant table 408 and a single global deny entry is made in the **deny table** 410.

Detailed Description Text - DETX (244):
1. Check the **Deny table** to see if the User U1 has a global deny (i.e., a deny to all objects). If so, check the Grant table to see if the user has specific granted items (objects) that are exceptions to the deny. If any such objects exist, grant access if the current operation matches the operation

specified in the Grant table, otherwise deny access.

Detailed Description Text - DETX (245):
2. Check the Grant table to see if the User U1 has a global grant (i.e. a grant to all objects). If so, check the **Deny table** to see if the user has specific denied items (objects) that are exceptions to the grant. If any such objects exist, deny access if the current operation matches the operation specified in the **Deny table**, otherwise grant access.

Detailed Description Text - DETX (246):
3. Check the **Deny table** to see if User U1 has specific denied items (objects), and deny access if the current operation matches the operation specified in the **Deny table**.

Detailed Description Text - DETX (248):
5. Check the **Deny table** to see if there is an all-users global deny (i.e., deny all objects to all users). If so, check the Grant table to see if all users have specific granted items (objects) that are exceptions to the deny. If any such objects exist, grant access if the current operation matches the operation specified in the Grant table, otherwise deny access.

Detailed Description Text - DETX (249):
6. Check the Grant table to see if there is an all-users global grant (i.e., grant all objects to all users). If so, check the **Deny table** to see if all users have specific denied items (objects) that are exceptions to the grant. If any such objects exist, deny access if the current operation matches the operation specified in the **Deny table**, otherwise grant access.

Detailed Description Text - DETX (250):
7. Check the **Deny table** to see if all users have specific denied items (objects). If so, deny access if the current operation matches the operation specified in the **Deny table**.

Detailed Description Text - DETX (256):
FIGS. 15A and 15B depict an exemplary Grant table and **Deny table**

respectively.

Detailed Description Text - DETX (294):
permission tables 406 including the grant table 408 and the **deny table** 410;

Detailed Description Paragraph Table - DETL (4):

GRANT TABLE: (U1, NULL, Op1)

DENY TABLE: (U1, X1, Op1) (U1, X2, Op1) (U1, X3, Op1)

Detailed Description Paragraph Table - DETL (5):

DENY TABLE: (U1, NULL, Op1)

GRANT TABLE: (U1, X1, Op1) (U1, X2, Op1) (U1, X3, Op1)

L Number	Hits	Search Text	DB	Time stamp
20	16928	(database or (data adj base)) near3 (monitor$4 or manag$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 11:59
21	783486	valid or secure	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 11:59
23	1999017	valid or secur$4	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 12:00
24	4210	(reject$4 or den$4) adj2 (list or table)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 12:01
25	1090	((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 12:02
26	0	(((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4)) same ((reject$4 or den$4) adj2 (list or table))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 12:02
27	22	(((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4)) and ((reject$4 or den$4) adj2 (list or table))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:02
28	380284	registry	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:01
29	2	6038563.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:02
30	1	registry and 6038563.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:02

31	1	(registry and 6038563.pn.) and (((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:03
32	571563	ID	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:03
33	559773	key	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:03
34	0	ID and ((registry and 6038563.pn.) and (((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:03
35	1	key and ((registry and 6038563.pn.) and (((database or (data adj base)) near3 (monitor$4 or manag$4)) same (valid or secur$4)))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:08
36	4442	key adj value	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:08
37	1553	process adj ID	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:09
38	0	registery adj key	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:09
39	28	leaking adj data	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:09
40	204655	database	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:09

41	351	registry adj key	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:10
42	0	(leaking adj data) and database and (registry adj key)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:10
43	3	(leaking adj data) and database	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/01/07 13:10

US-PAT-NO: 6553466

DOCUMENT-IDENTIFIER: US 6553466 B1

TITLE: Shared memory blocking method and system

DATE-ISSUED: April 22, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP	CODE	COUNTRY
Friedman; George	Austin	TX	N/A		N/A
Starek; Robert Phillip	Austin	TX	N/A		N/A
Murdock; Carlos A.	Austin	TX	N/A		N/A

US-CL-CURRENT: 711/152, 707/8 , 711/163

ABSTRACT:

A shared memory blocking method and particularly applicable to a system in which protected data is transmitted to a recipient computer. The method comprises reserving a memory page for a requesting application, committing a memory page to the requesting application's address space, which call may be made by the process providing the page reserve call or by a subsequent process, and providing security checks to complete the requests. The security checks include determining whether the process is secured by consulting a secured process list and determining whether the page is shared by consulting a shared memory list. Further disclosed are a computer readable medium and computer programmed to block shared memory, shared memory blocking system and secured data transmission system.

25 Claims, 5 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

---------- KWIC ---------

Brief Summary Text - BSTX (5):

Shared memory may be exploited for **leaking data**. "**Leaking data**" as used herein means transferring data out of a system in which it is desired to have the data secured. A data leak may occur if a process writes information to a shared memory location and another process accesses the information from that location.

Detailed Description Text - DETX (6):

A package carries data and provides associated information to a command center which is a component of an application programming interface, such as a Win32 process. A communication driver handles communication between the application programming interface and a plurality of device drivers. It provides a single set of device driver I/O control functions that are called from the application programming interface to send information to or retrieve information from the device drivers. The communication driver is called by a hook driver to notify the command center that a process is trying to open a packaged file. The device drivers, together with the application programming interface, marshal the packaged content into a vault and support access to the content, subject to an originator's permission selection. The command center may watch for packages to be executed and prompt users for file names to save a package payload. It may notify the file system hook driver that a package payload should be absorbed into the vault. It may present users with dialog indicating that an application is attempting to open a packaged file. It may also notify device drivers 160 when applications exit. The command center may block clipboard access and terminate applications at the request of a permissions device driver when permissions expire. Permission information is contained in a **database** and may include, for example, file names, package ID, file system ID and file permissions. File permissions may include, but are not limited to, length of time or number of times a file may be open, date after which a file may no longer be opened, and printing and clipboard permissions.